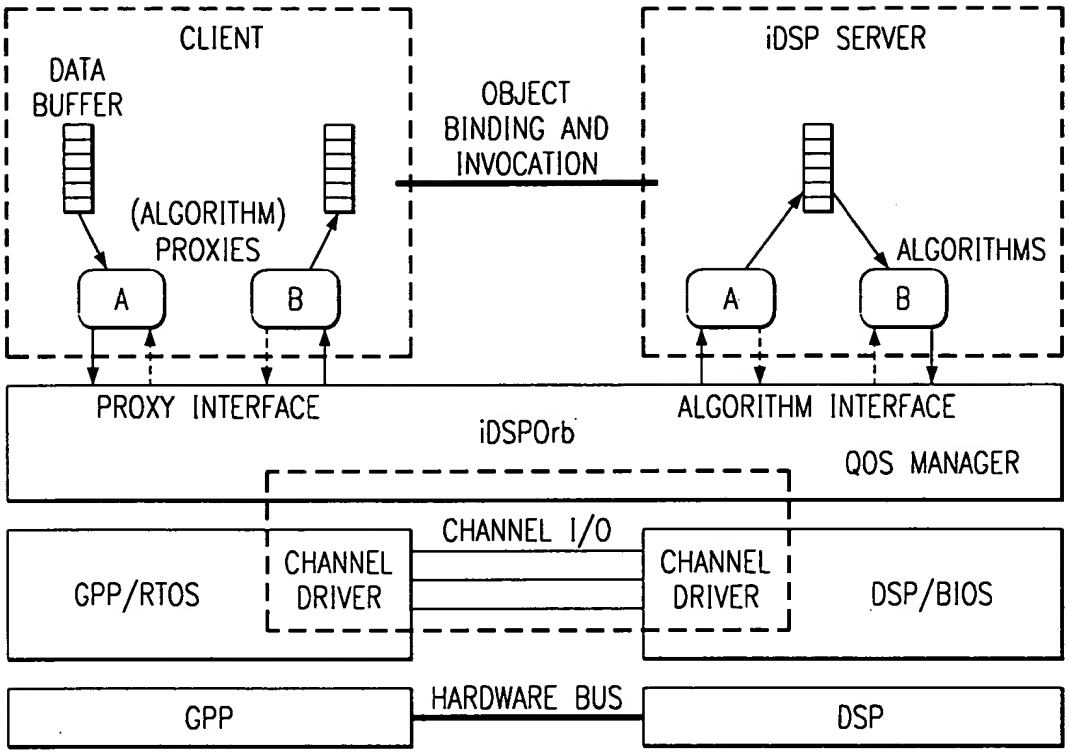


FIG. 1

FIG. 2

ALGORITHM DSPIDL

```
module H263{
  interface TIDEC{
      int decode([in] STREAM,
                 [out] STREAM);
  }
}
```

DSPIDL COMPILER

PROXY INTERFACE

```
H263_TIDEC*
H263_TIDEC_create(
  DSPORB_Params* p);

int
H263_TIDEC_decode(H263_TIDEC* d,
          DSPORB_Stream* in,
          DSPORB_Stream* out);
```

ALGORITHM INTERFACE

```
H263_TIDEC_Handle
H263_TIDEC_create(IALG_Params p);

int
H263_TIDEC_decode(
          H263_TIDEC_Handle h,
          SIO_Handle in,
          SIO_Handle out);
```

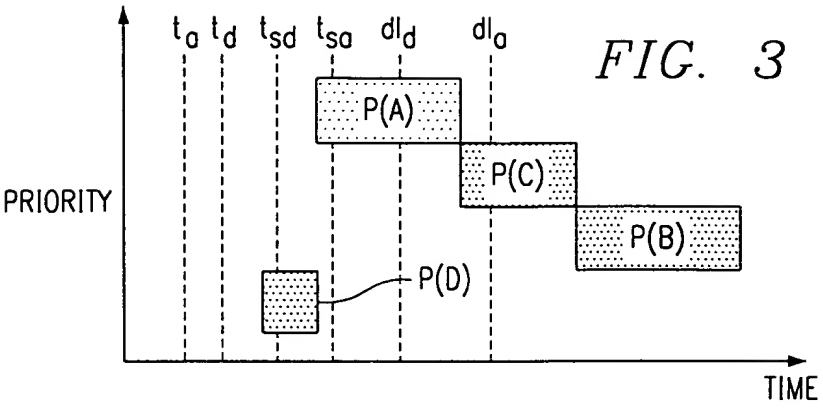


FIG. 3

t_{sa} = LAST POSSIBLE TIME FOR PROCESS A TO START AND STILL MAKES ITS DEADLINE

t_{sd} = LAST POSSIBLE TIME FOR PROCESS D TO START AND STILL MAKE ITS DEADLINE

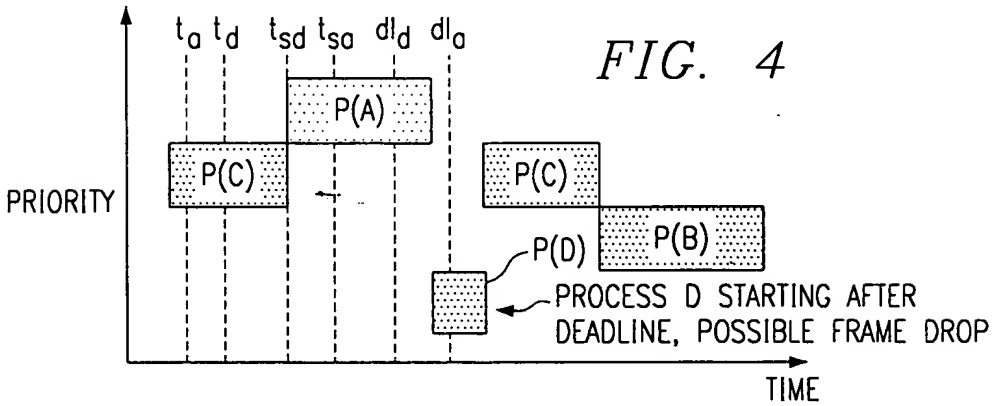


FIG. 4

t_{sa} = LAST POSSIBLE TIME FOR PROCESS A TO START AND STILL MAKES ITS DEADLINE

t_{sd} = LAST POSSIBLE TIME FOR PROCESS D TO START AND STILL MAKE ITS DEADLINE

FIG. 5

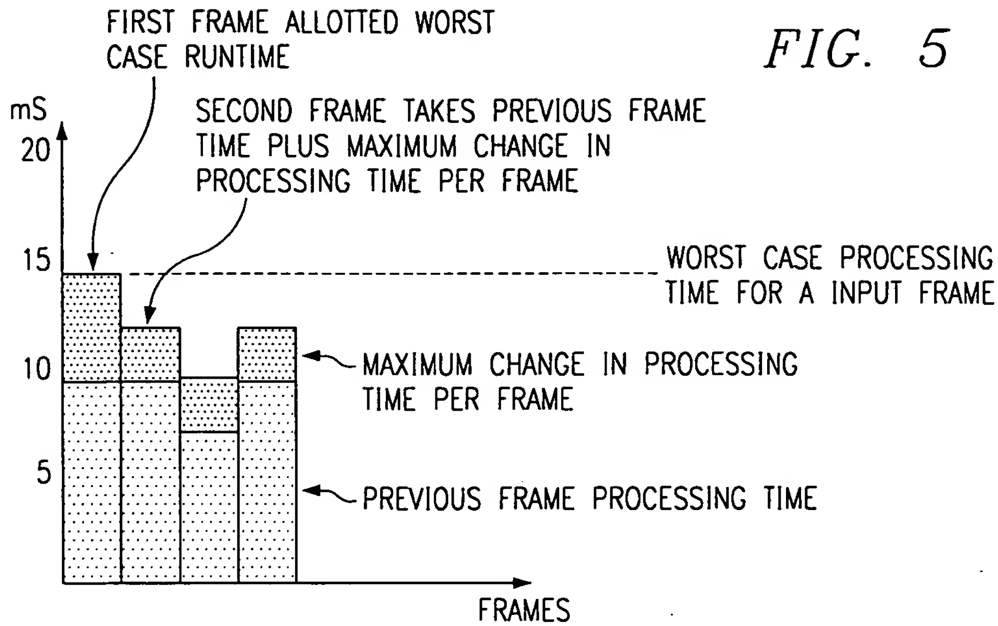

FIRST FRAME ALLOTTED WORST CASE RUNTIME
mS
20
SECOND FRAME TAKES PREVIOUS FRAME TIME PLUS MAXIMUM CHANGE IN PROCESSING TIME PER FRAME
15
WORST CASE PROCESSING TIME FOR A INPUT FRAME
10
MAXIMUM CHANGE IN PROCESSING TIME PER FRAME
5
PREVIOUS FRAME PROCESSING TIME
FRAMES

FIG. 6

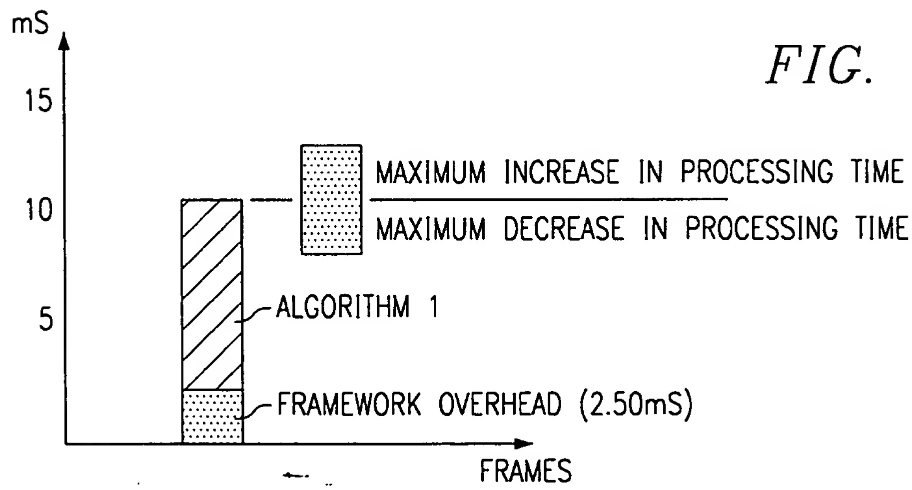


FIG. 7

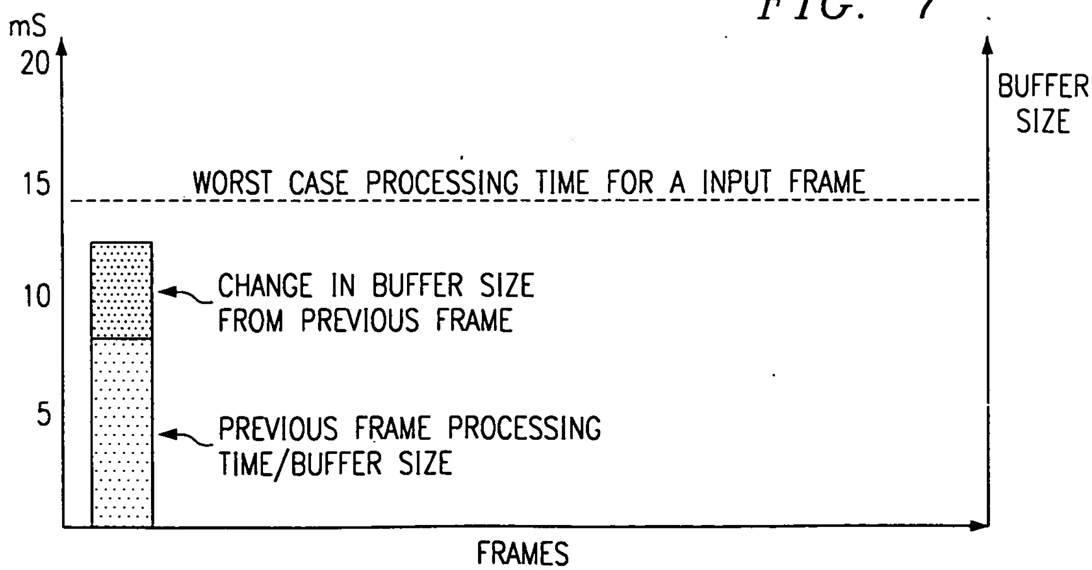


FIG. 8

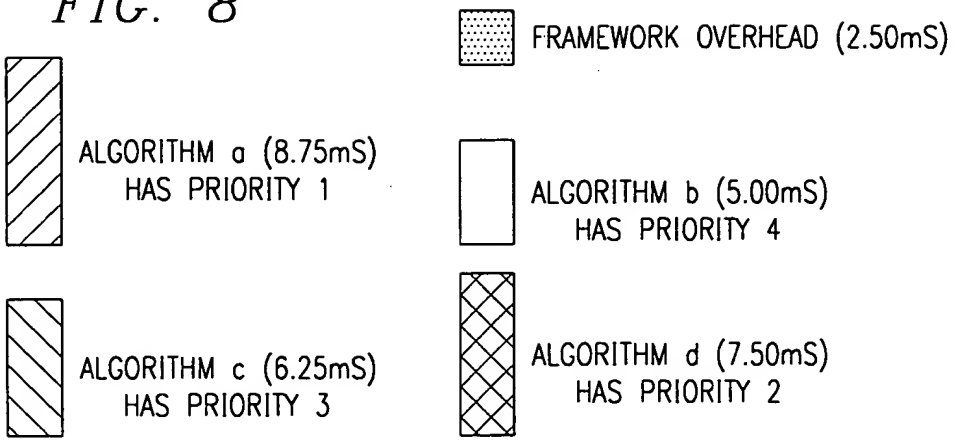
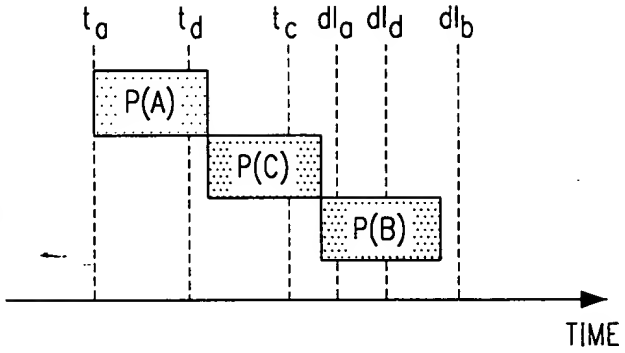


FIG. 9

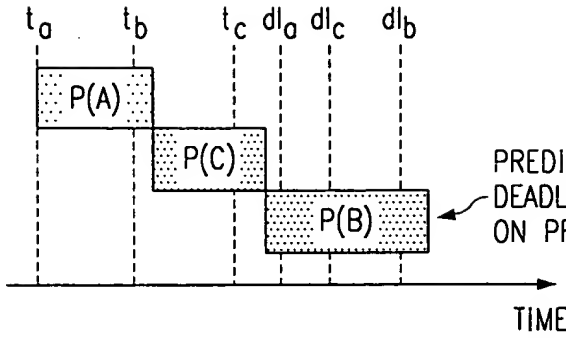


t_i = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS

dl_i = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME

P() = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME

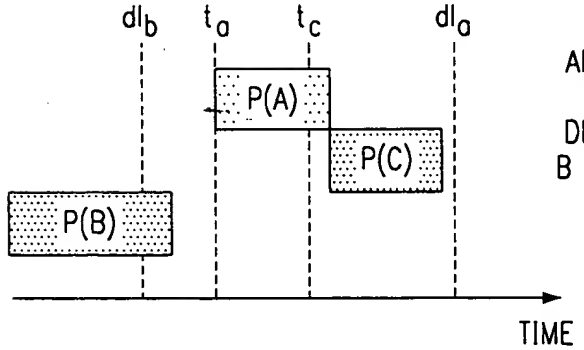
FIG. 10



t_i = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS

dl_i = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME

P() = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME



BOTH PROCESS A AND C ARE PREDICTED TO COMPLETE BEFORE THEIR RESPECTIVE DEADLINES MEANING PROCESS B MISSING ITS DEADLINE DOES NOT RIPPLE THROUGH THE SYSTEM (YET)

t_i = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS

dl_i = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME

P() = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME

FIG. 11

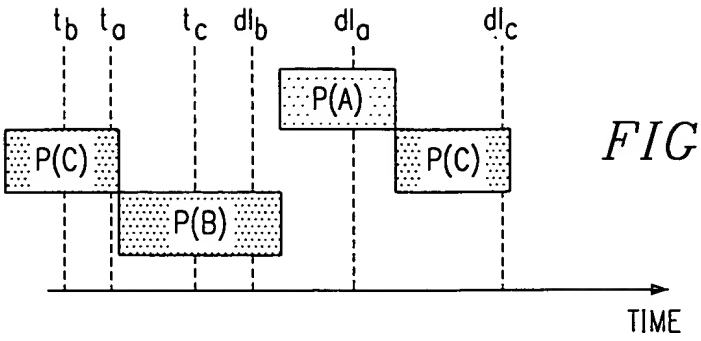


FIG. 12

t_i = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS

dl_i = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME

P() = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME

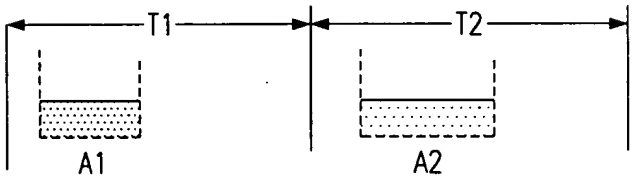


FIG. 13a

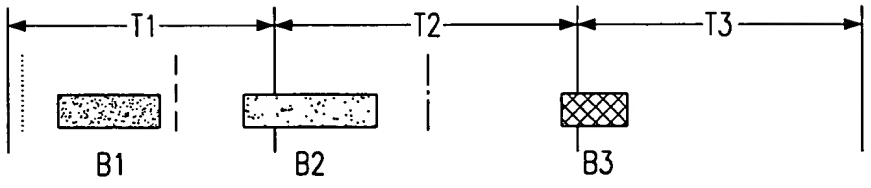


FIG. 13b

..... ARRIVAL OF BUFFER B1

- - - ARRIVAL OF BUFFER B2

—·— ARRIVAL OF BUFFER B3

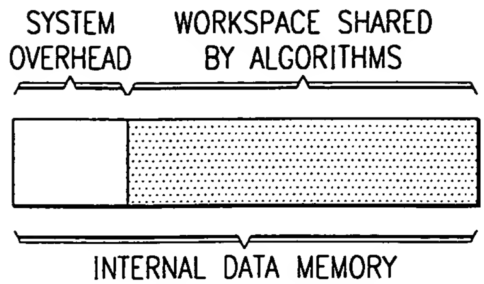


FIG. 14

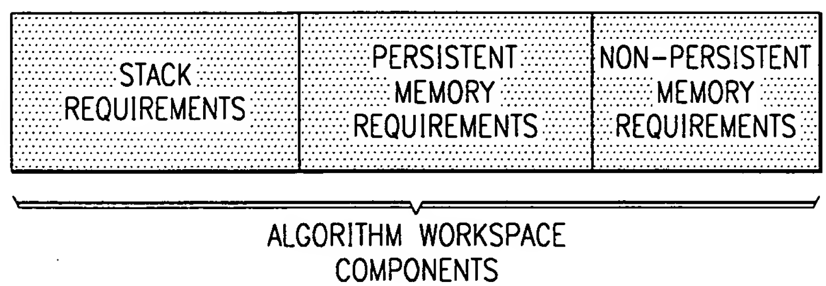


FIG. 15

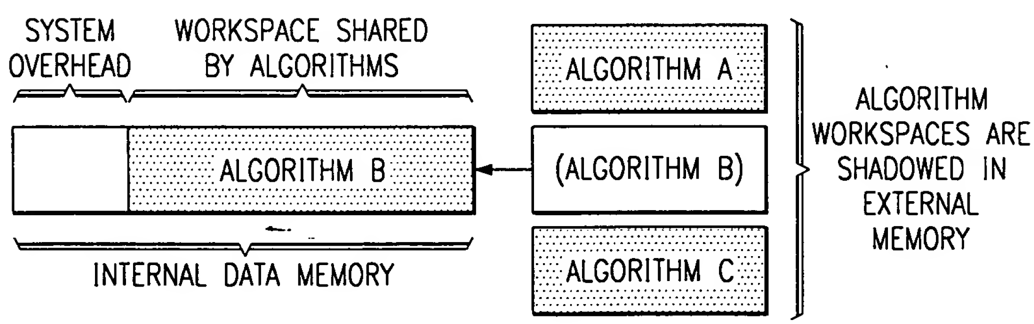


FIG. 16

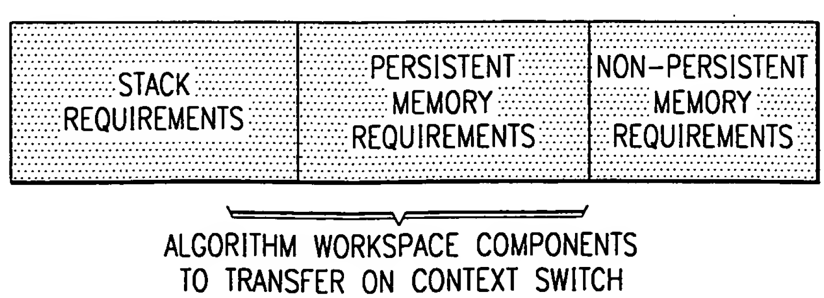


FIG. 17

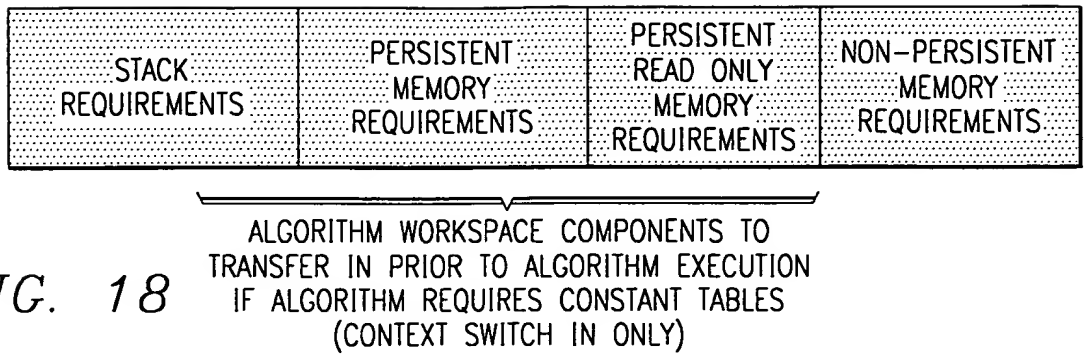


FIG. 18

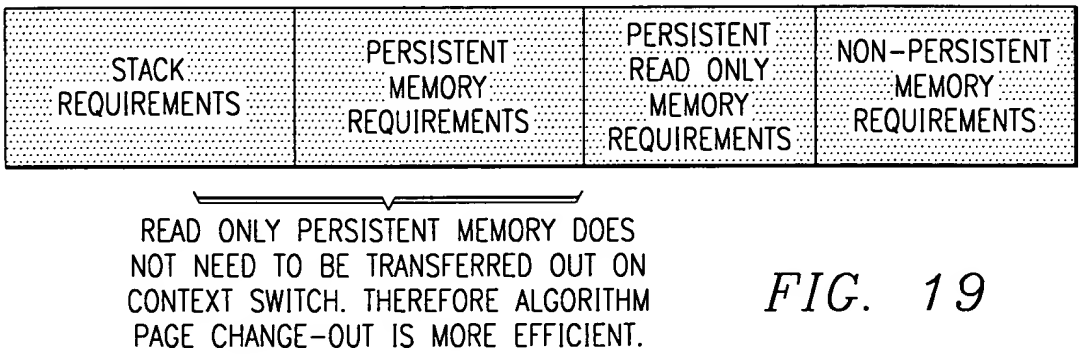


FIG. 19

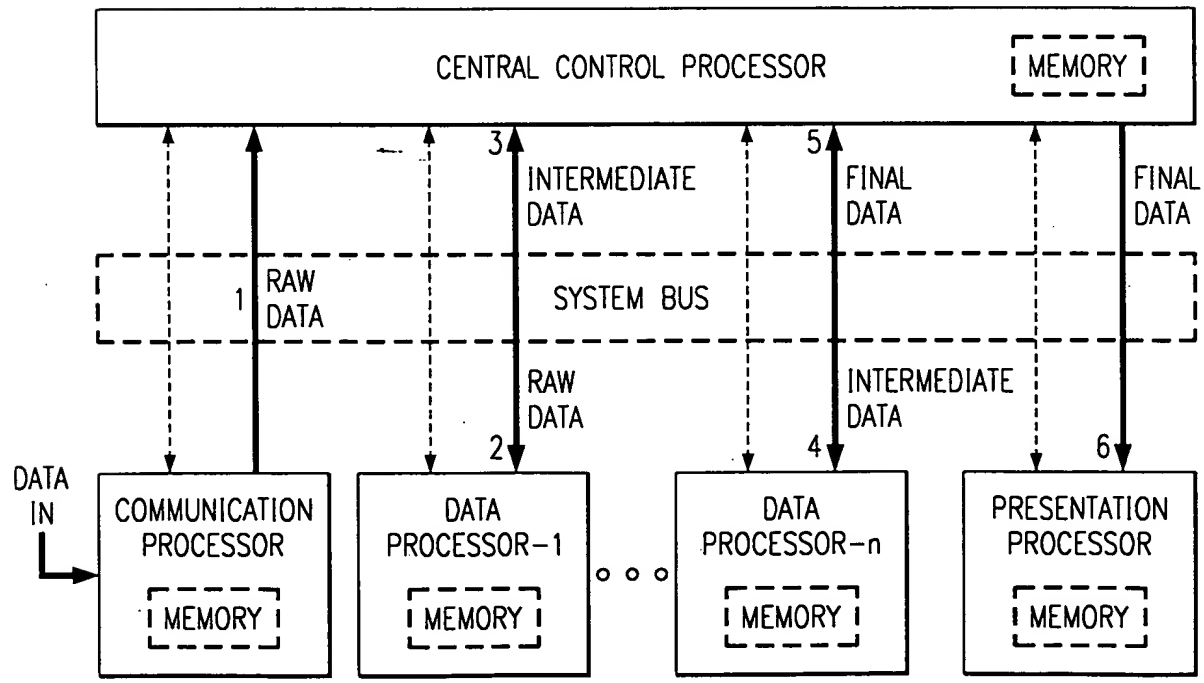
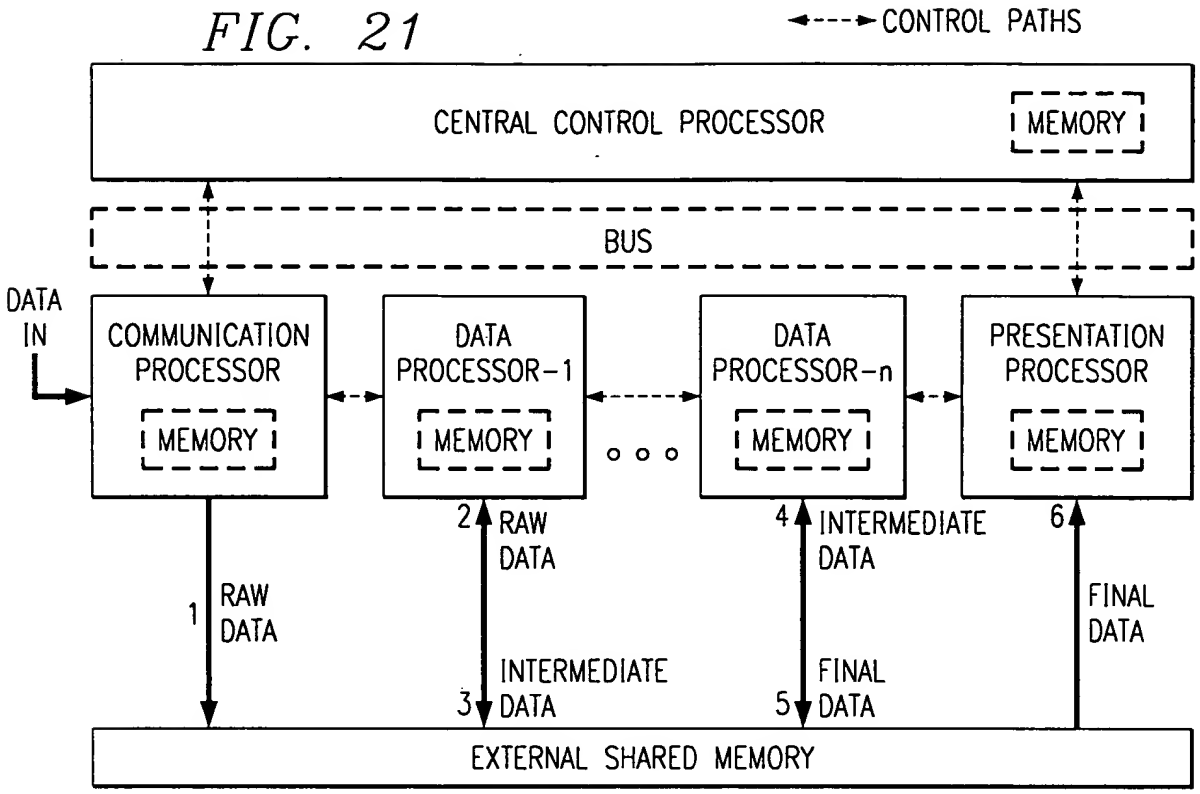


FIG. 20
(PRIOR ART)

FIG. 21


CONTROL PATHS
CENTRAL CONTROL PROCESSOR
MEMORY
BUS
DATA IN
COMMUNICATION PROCESSOR
MEMORY
DATA PROCESSOR-1
MEMORY
DATA PROCESSOR-n
MEMORY
PRESENTATION PROCESSOR
MEMORY
1 RAW DATA
2 RAW DATA
3 INTERMEDIATE DATA
4 INTERMEDIATE DATA
5 FINAL DATA
6 FINAL DATA
EXTERNAL SHARED MEMORY

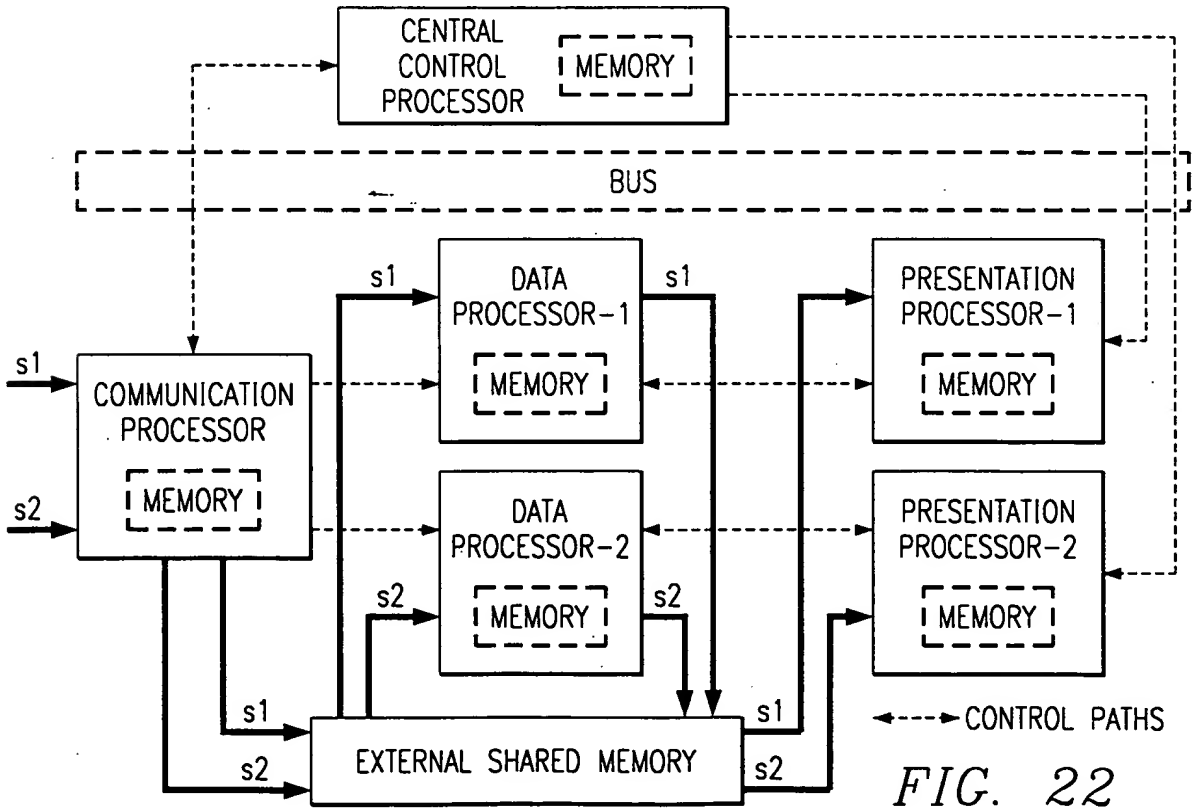
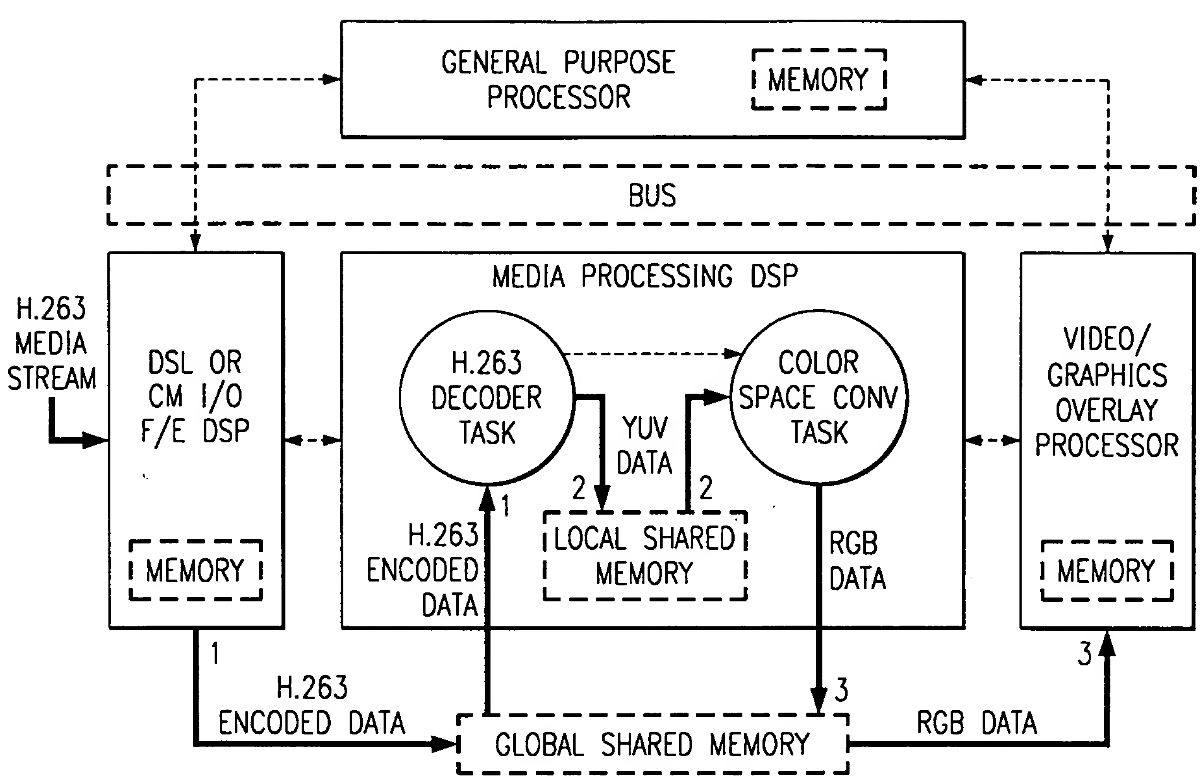

CENTRAL CONTROL PROCESSOR
MEMORY
BUS
s1
s2
COMMUNICATION PROCESSOR
MEMORY
DATA PROCESSOR-1
MEMORY
DATA PROCESSOR-2
MEMORY
PRESENTATION PROCESSOR-1
MEMORY
PRESENTATION PROCESSOR-2
MEMORY
EXTERNAL SHARED MEMORY
CONTROL PATHS

FIG. 22

FIG. 23


CONTROL PATHS
GENERAL PURPOSE PROCESSOR
MEMORY
BUS
H.263 MEDIA STREAM
DSL OR CM I/O F/E DSP
MEMORY
MEDIA PROCESSING DSP
H.263 DECODER TASK
YUV DATA
COLOR SPACE CONV TASK
H.263 ENCODED DATA
LOCAL SHARED MEMORY
RGB DATA
VIDEO/ GRAPHICS OVERLAY PROCESSOR
MEMORY
1
2
2
1
3
3
H.263 ENCODED DATA
GLOBAL SHARED MEMORY
RGB DATA

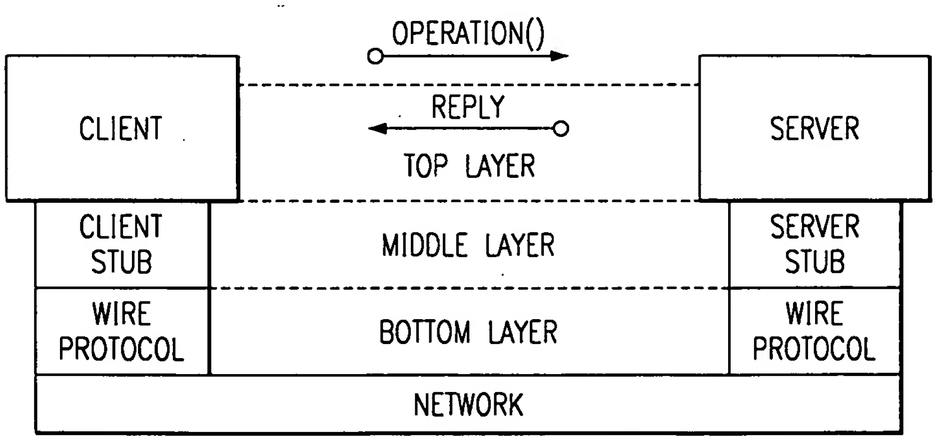

OPERATION()
CLIENT
REPLY
TOP LAYER
SERVER
CLIENT STUB
MIDDLE LAYER
SERVER STUB
WIRE PROTOCOL
BOTTOM LAYER
WIRE PROTOCOL
NETWORK

FIG. 24

FIG. 25

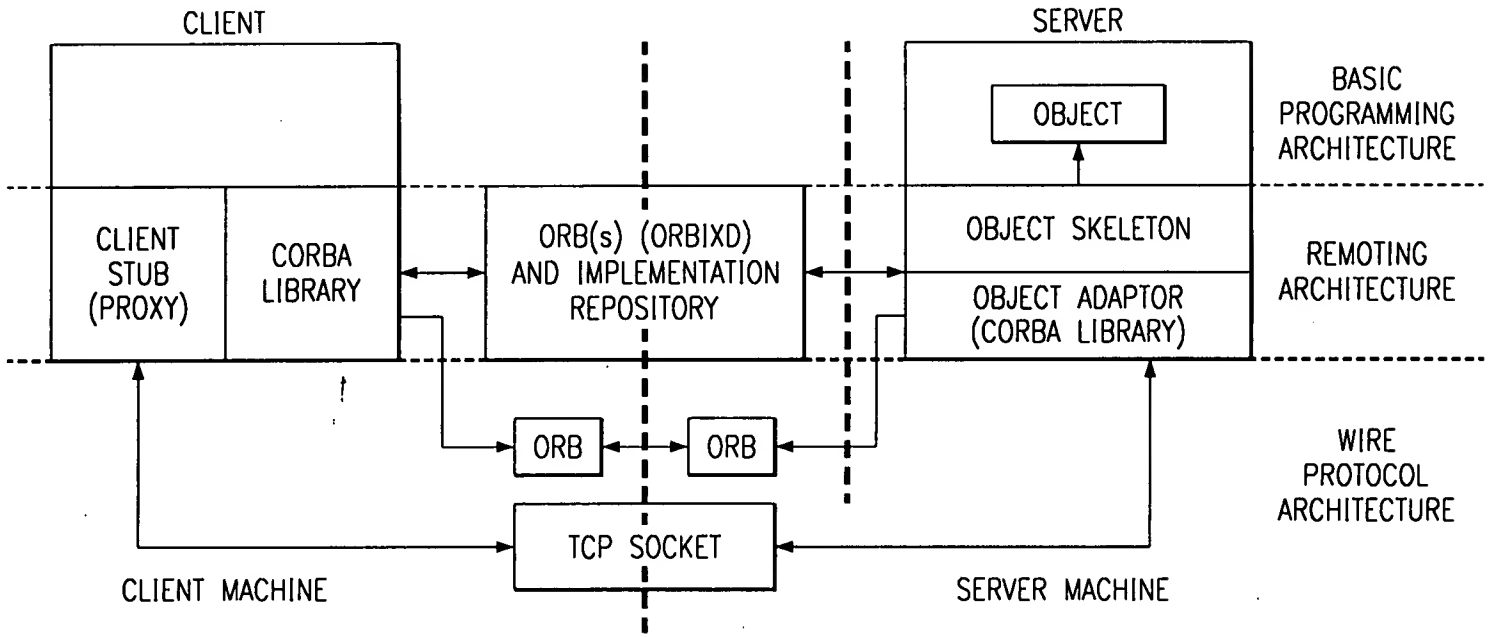


FIG. 26a

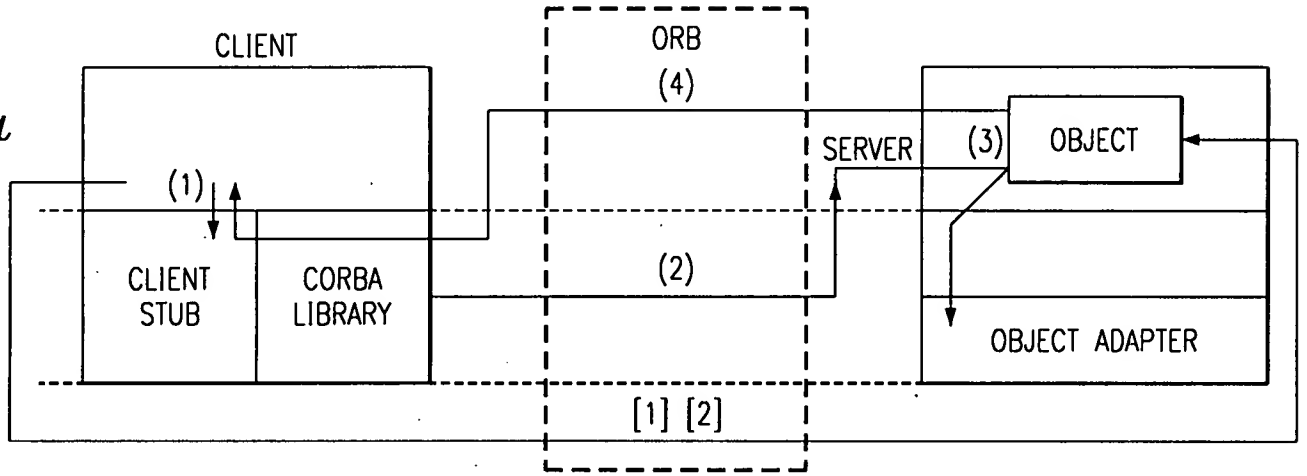


FIG. 26b

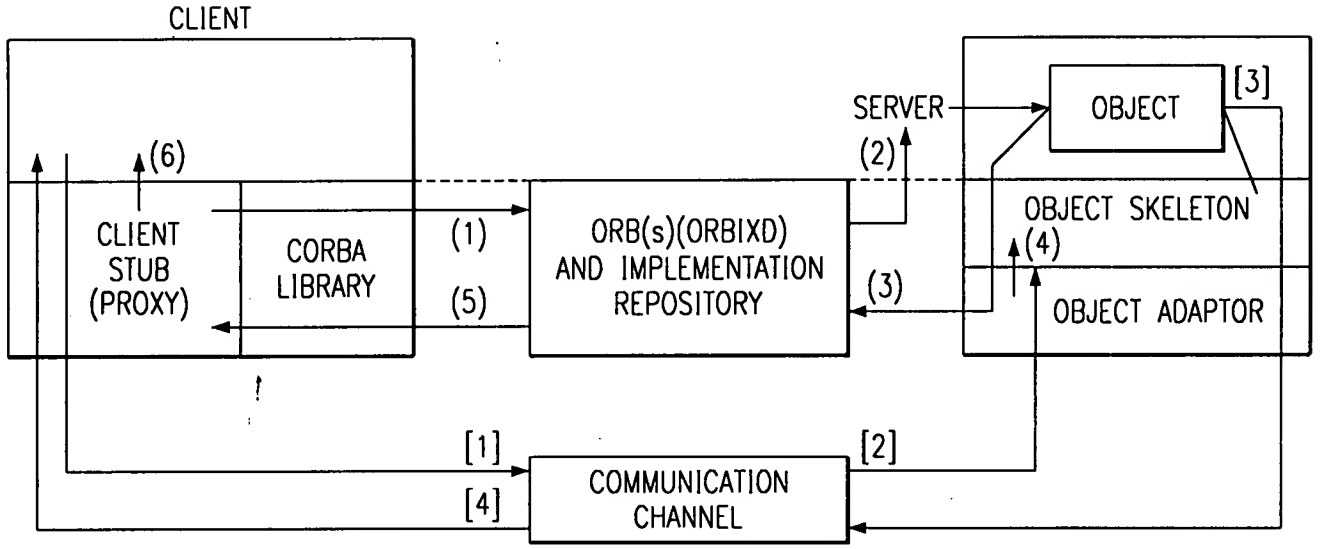
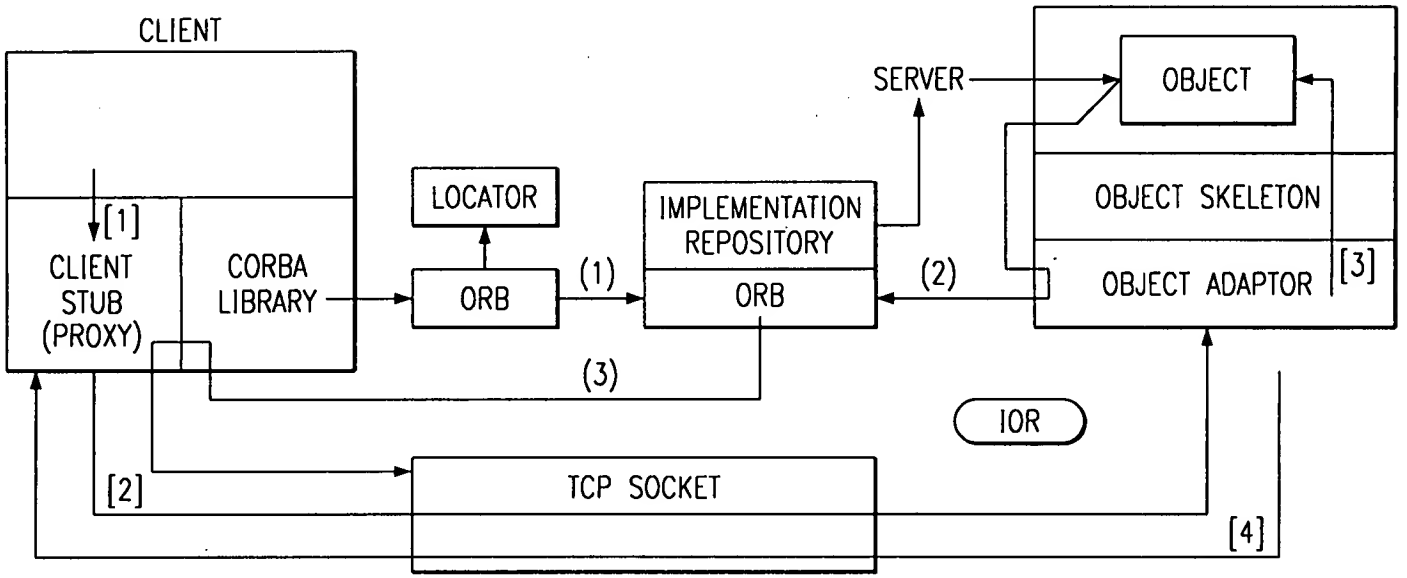


FIG. 26c



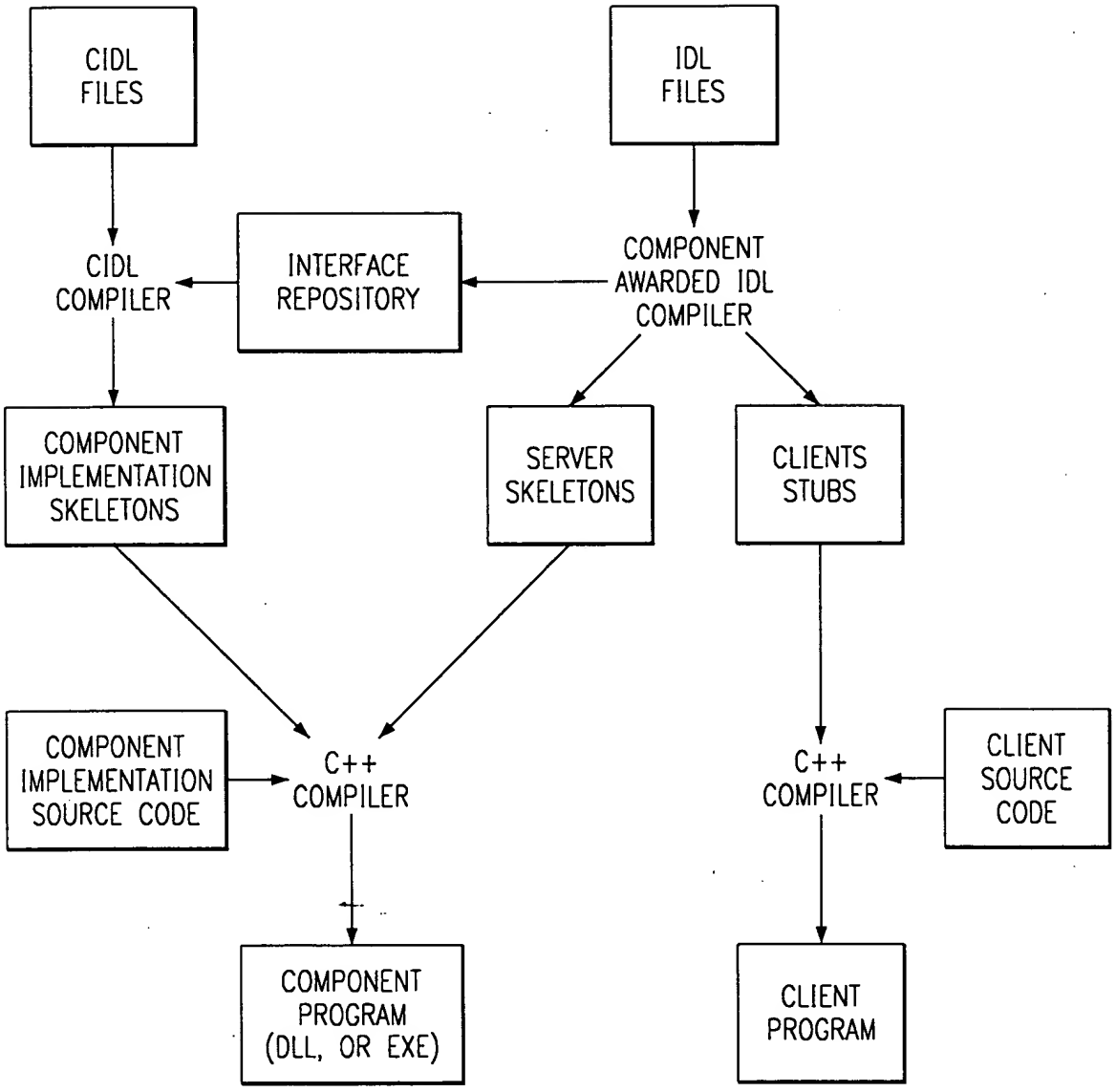

CIDL FILES
IDL FILES
CIDL COMPILER
INTERFACE REPOSITORY
COMPONENT AWARDED IDL COMPILER
COMPONENT IMPLEMENTATION SKELETONS
SERVER SKELETONS
CLIENTS STUBS
COMPONENT IMPLEMENTATION SOURCE CODE
C++ COMPILER
C++ COMPILER
CLIENT SOURCE CODE
COMPONENT PROGRAM (DLL, OR EXE)
CLIENT PROGRAM

FIG. 27